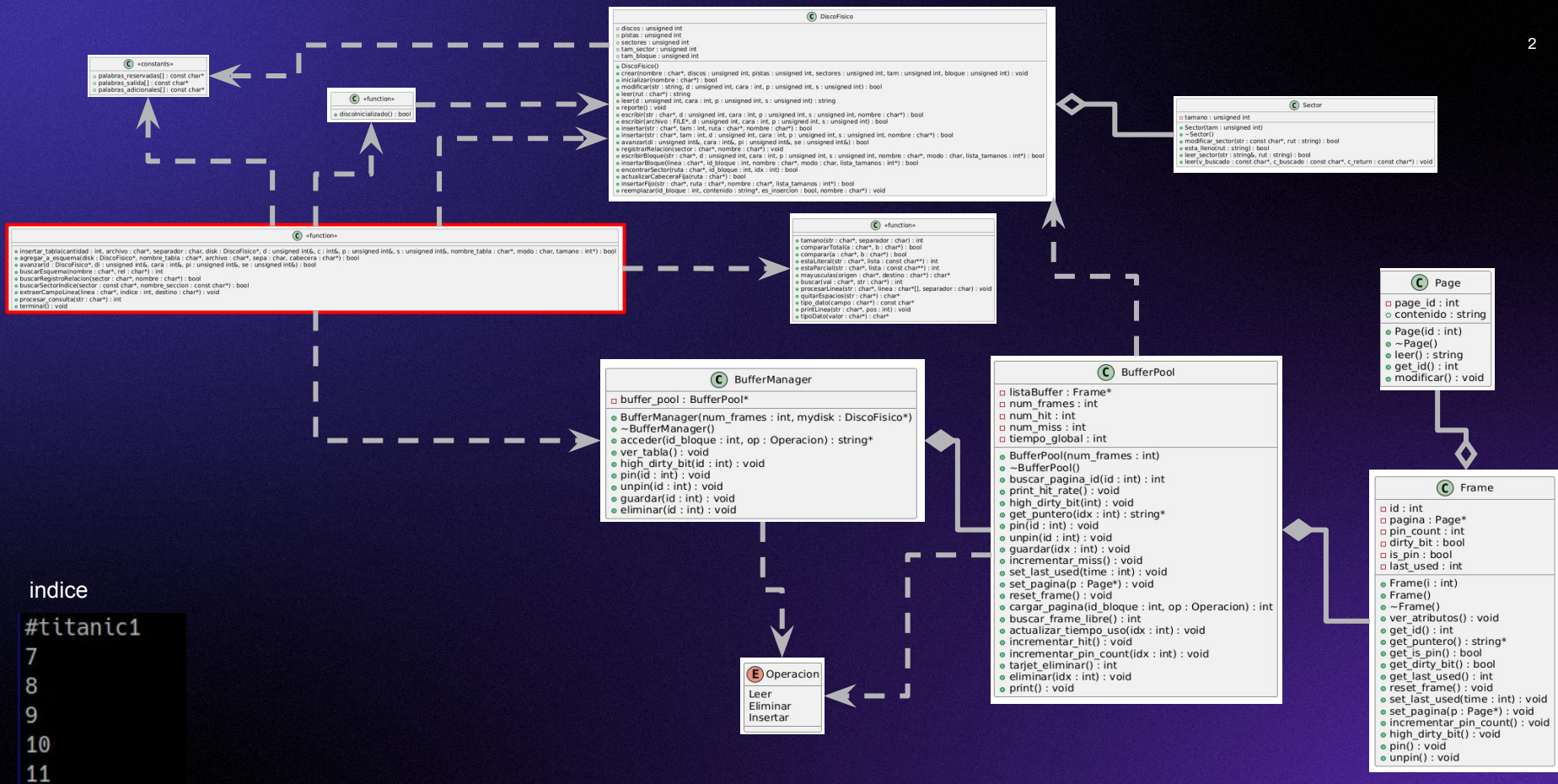


# Buffer Manager

By Jesus Alfaro Buiza

Diagrama UML  
POO para Buffer Manager  
Política LRU  
Capturas de ejecución



# Ejemplo Rápido

Bloques que accederemos

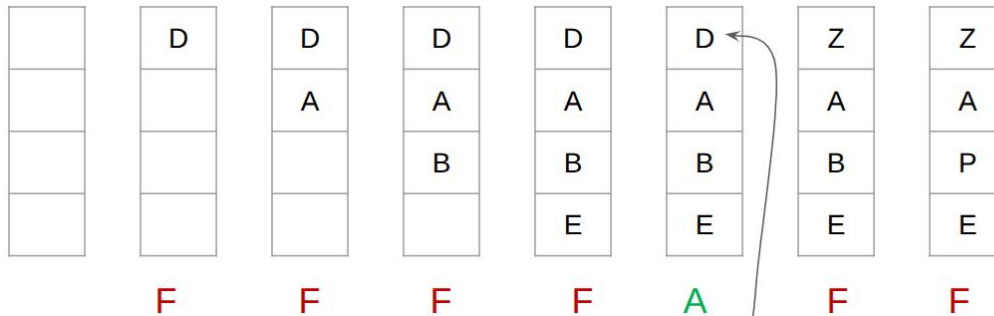
D	A	B	E	A	Z	P
---	---	---	---	---	---	---

Casos al acceder una página en Buffer Pool:

Acierto (hit)

Fallo (miss)

Buffer Pool



# Aciertos = 1

# Fallos = 6

Hit rate =  $1/(6+1)$   
=14.29%

Tablas con sus respectivos bloques

Titanic: D
Tabla 2: A
Tabla 3: B,E
Tabla 4 : Z,P

■ Liberamos/eliminamos D

verifica el Dirty-bit

- Si se modificó D, se escribe en el disco
- Si solo fue lectura simplemente se elimina



### C BufferManager

□ buffer\_pool : BufferPool\*

● BufferManager(num\_frames : int, mydisk : DiscoFisico\*)  
● ~BufferManager()  
● acceder(id\_bloque : int, op : Operacion) : string\*  
● ver\_tabla() : void  
● high\_dirty\_bit(id : int) : void  
● pin(id : int) : void  
● unpin(id : int) : void  
● guardar(id : int) : void  
● eliminar(id : int) : void

### C BufferPool

□ listaBuffer : Frame\*  
□ num\_frames : int  
□ num\_hit : int  
□ num\_miss : int  
□ tiempo\_global : int

● BufferPool(num\_frames : int)  
● ~BufferPool()  
● buscar\_pagina\_id(id : int) : int  
● print\_hit\_rate() : void  
● high\_dirty\_bit(int) : void  
● get\_puntero(idx : int) : string\*  
● pin(id : int) : void  
● unpin(id : int) : void  
● guardar(idx : int) : void  
● incrementar\_miss() : void  
● set\_last\_used(time : int) : void  
● set\_pagina(p : Page\*) : void  
● reset\_frame() : void  
● cargar\_pagina(id\_bloque : int, op : Operacion) : int  
● buscar\_frame\_libre() : int  
● actualizar\_tiempo\_uso(idx : int) : void  
● incrementar\_hit() : void  
● incrementar\_pin\_count(idx : int) : void  
● tarjet\_eliminar() : int  
● eliminar(idx : int) : void  
● print() : void

### C Frame

□ id : int  
□ pagina : Page\*  
□ pin\_count : int  
□ dirty\_bit : bool  
□ is\_pin : bool  
□ last\_used : int

● Frame(i : int)  
● Frame()  
● ~Frame()  
● ver\_atributos() : void  
● get\_id() : int  
● get\_puntero() : string\*  
● get\_is\_pin() : bool  
● get\_dirty\_bit() : bool  
● get\_last\_used() : int  
● reset\_frame() : void  
● set\_last\_used(time : int) : void  
● set\_pagina(p : Page\*) : void  
● incrementar\_pin\_count() : void  
● high\_dirty\_bit() : void  
● pin() : void  
● unpin() : void

### E Operacion

Leer  
Eliminar  
Insertar

### C Page

□ page\_id : int  
○ contenido : string

● Page(id : int)  
● ~Page()  
● leer() : string  
● get\_id() : int  
● modificar() : void

FrameId	PageId	Dirty Bit	is Pin	Pin Count	Last Used
0	5	false	false	1	6
1	2	false	false	2	3
2	3	false	false	1	4
3	4	false	false	1	5

#hits = 1

#miss = 5

hit\_rate = 0.17

pagina inicializada con contenido

pagina cargada

## TABLA DE METADATOS DEL BUFFER POOL

```
int BufferPool::tarjet_eliminar(){
    int lru_idx = -1;
    int min_time;
    bool first_found = false;

    for (int i = 0; i < num_frames; i++){
        if(!listaBuffer[i].get_is_pin()){
            if(!first_found){
                min_time = listaBuffer[i].get_last_used();
                lru_idx = i;
                first_found = true;
            } else if(listaBuffer[i].get_last_used() < min_time){
                min_time = listaBuffer[i].get_last_used();
                lru_idx = i;
            }
        }
    }
    return lru_idx;
}
```

```
void BufferPool::eliminar(int idx){
    guardar(idx);
    listaBuffer[idx].reset_frame();
}
```

```
void BufferPool::guardar(int idx){
    if(listaBuffer[idx].get_dirty_bit()){
        int opcion;
        printf("desea guardar cambios? de %d\n si == 1\n no== 0\nopcion: ", listaBuffer[idx].get_id());
        scanf("%d",&opcion);
        getchar();
        if (opcion){
            discoUsado->reemplazar(listaBuffer[idx].get_id(), listaBuffer[idx].get_puntero());
            printf("bloque %d actualizado\n", listaBuffer[idx].get_id());
        }
    }
}
```

## Polica de reemplazo LRU

Se elimina el menos reciente, pero se considera si está fijado (pin)



# Ejemplo Rápido

7

Bloques que accederemos

D	A	B	E	A	Z	P
M	M	L	M	M		

Operaciones:

- Seleccionar
- Eliminar
- Insertar

Buffer Pool

	D	D	D	D	Z
		A	A	A	A
			B	B	P
			E	E	E

Fijación (Pin)

Bloqueo Exclusivo

Bloqueo Compartido

Tablas con sus respectivos bloques

Titanic: D
Tabla 2: A
Tabla 3: B,E
Tabla 4 : Z,P

## Contenido Paralelo en Buffer y Disco??

### Comandos:

#### Set-buffer

Permite configurar la cantidad de frames en nuestro buffer pool.

#### Bloque-disco 15

Muestra contenido en el contenido del bloque 15 (en disco)

#### Bloque-buffer 15 L/W

Muestra contenido en el contenido de la página 15 (en el buffer)

#### Guardar 15

Permite guardar el contenido en disco (realiza pregunta de verificación)

#### Eliminar 15

Elimina la página 15 del buffer pool, aplica guardar (solo si es dirty-bit esta activo)

#### Buffer

Muestra la tabla de metadatos del buffer pool.



## Contenido Paralelo en Buffer y Disco??

### Comandos:

#### **Bloque-buffer-pin 15 L**

Fija permanente el bloque 15 (solo sirve para accesos manuales)

#### **Bloque-buffer-unpin 15 L**

Quitar el fija permanente del bloque 15 (solo sirve para accesos manuales)

